Deformable Contours

Prof. Dr. Elli Angelopoulou

Pattern Recognition Lab (Computer Science 5)

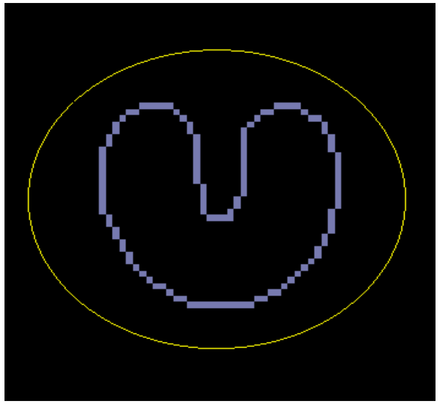
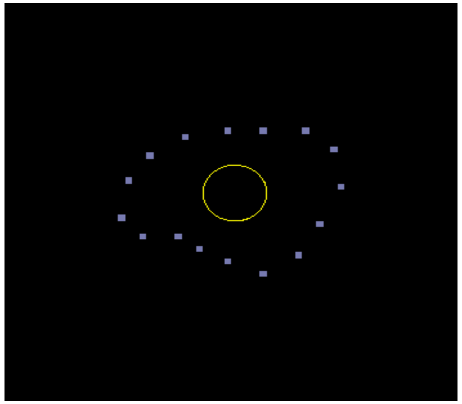
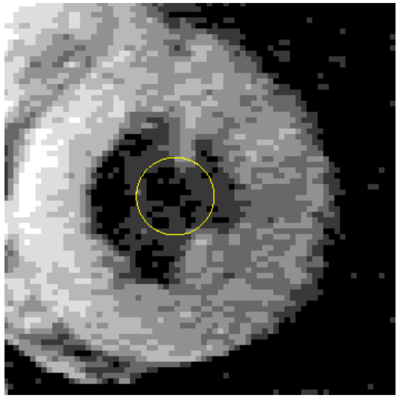
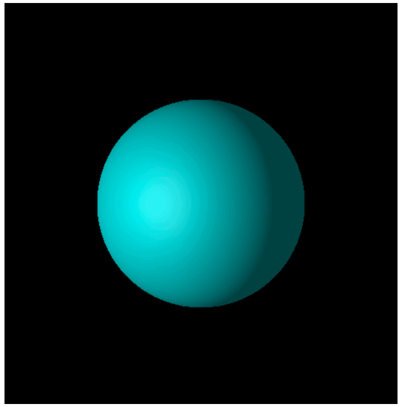
University of Erlangen-Nuremberg

Geometric Features

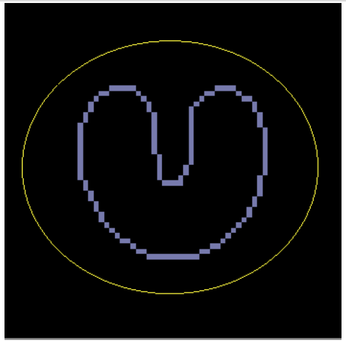
- We examined features that can be extracted directly from images:
  - Edges
  - Textons
  - Color
- We also examined the extraction of higher level features that correspond to specific shapes.
  - Lines
  - Circles
  - Ellipses
- Hough Transforms are well-suited for this last set of features. They can also be used for arbitrary shapes (Generalized Hough Transform) but this typically requires a considerable amount of pre-processing.
- Is there a better way to find curves of arbitrary shapes?

Deformable Contours

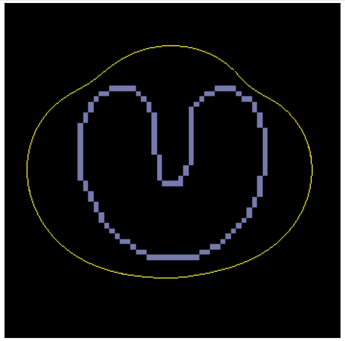
- Deformable contours are also known as active contours or snakes.
- Goal: find a contour that best approximates the perimeter of an object.
- One can visualize it as a rubber band of arbitrary shape that is capable of deforming during time, in order to get as close as possible to the target contour.

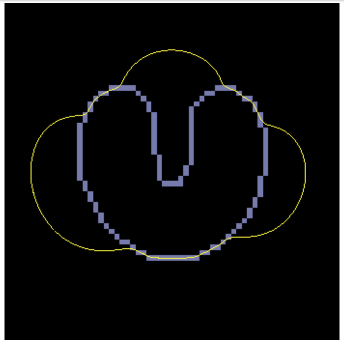
Deformable Contour Example



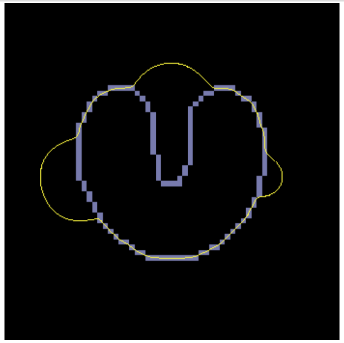
Initialization



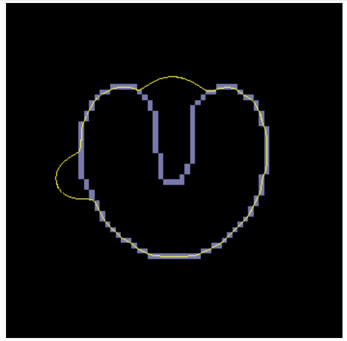
Iteration 2



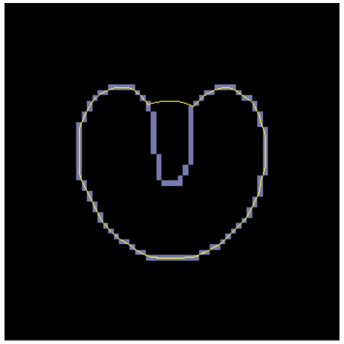
Iteration 4



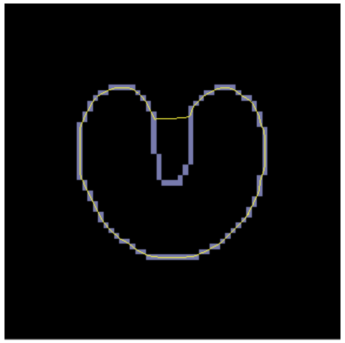
Iteration 7



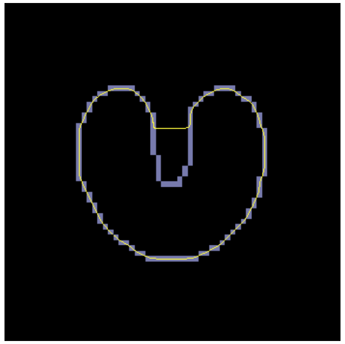
Iteration 9



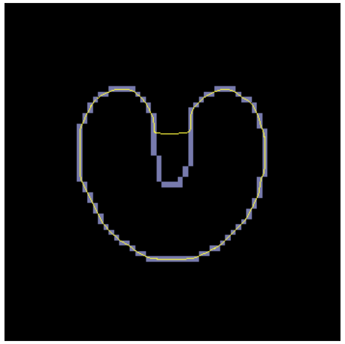
Iteration 11



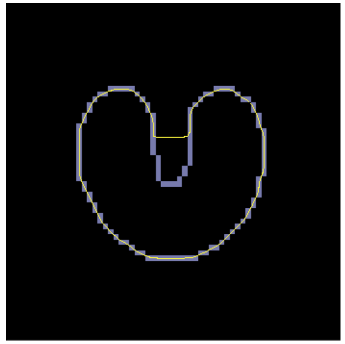
Iteration 13



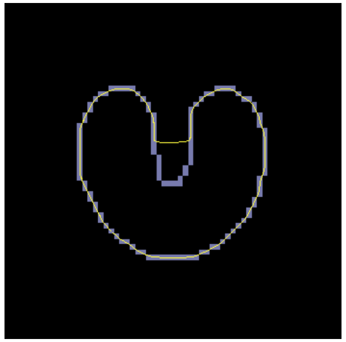
Iteration 16



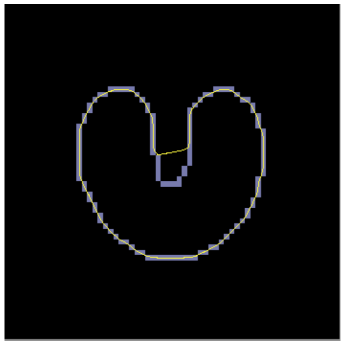
Iteration 18



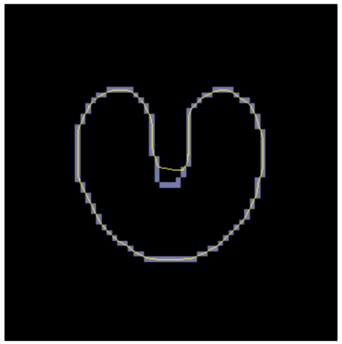
Iteration 20



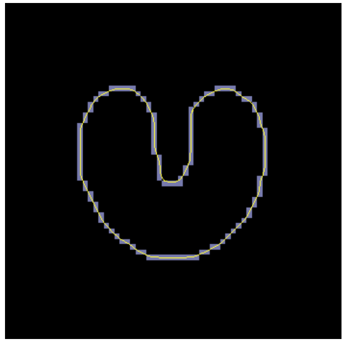
Iteration 22



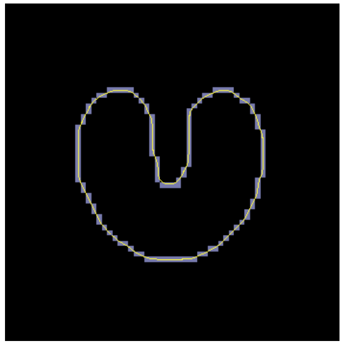
Iteration 24



Iteration 26



Iteration 28



Iteration 30

Main Idea of Deformable Contours

- The image information (usually edges) guide an elastic band.
- The rubber band is deformed, pulled, by the edges (or other image information) to fit the
- The band is located initially near the image contour of interest
- It is attracted toward the target contour by forces that depend on the intensity gradient.

Idea of Deformable Contours

- The image information (usually edges) guide an elastic band that is *sensitive to* the intensity gradient (or some other image feature).
- The band is initially located near the image contour of interest.
- The rubber band is deformed, pulled, by the edges (or other image information) to fit the target contour.
- The edge-based deformable contours explicitly use the intensity *gradient* of the image, unlike the Hough transform which is often based on only the existence of edge points.

Procedure

1. A contour (open or closed) is placed near the image contour of interest.

- The initial placement can be done manually or be the output of some other algorithm.
- “Seeding” the snake (step 1) can be critical in the success of finding the contour.

2. During an iterative process, the active contour is attracted towards the target contour by various forces that control the *shape and location of the snake within* the image.

"Pulling" Concept

- How is this band attracted to the target contour?
- We have to describe the forces that act on the contour to deform it.
- Different deformable contour models use different forces.
- We will cover the more classical formulation which is:
  - Based on intensity gradients
  - Given as a sum of 3 forces.

"Pulling" Forces

- The 3-forces active contour model uses the following three deformation-guiding forces:

1. A *continuity term* (force), E_{cont} which encourages continuity of the contour.
2. A *smoothness term* (force), E_{curv} which encourages smoothness in the contour.
3. An *edge attraction term* (force), E_{img} which pulls the contour towards the closest image edge.

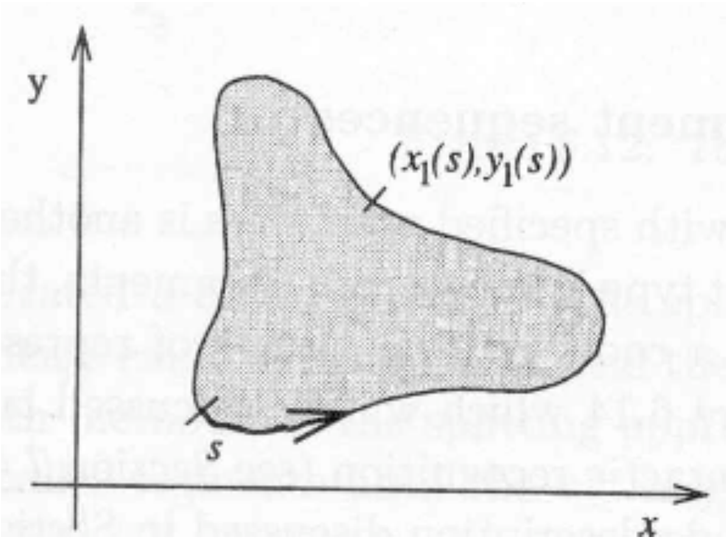
- E_{cont} and E_{curv} are called *internal energy* terms.
- E_{img} is called *external energy* term

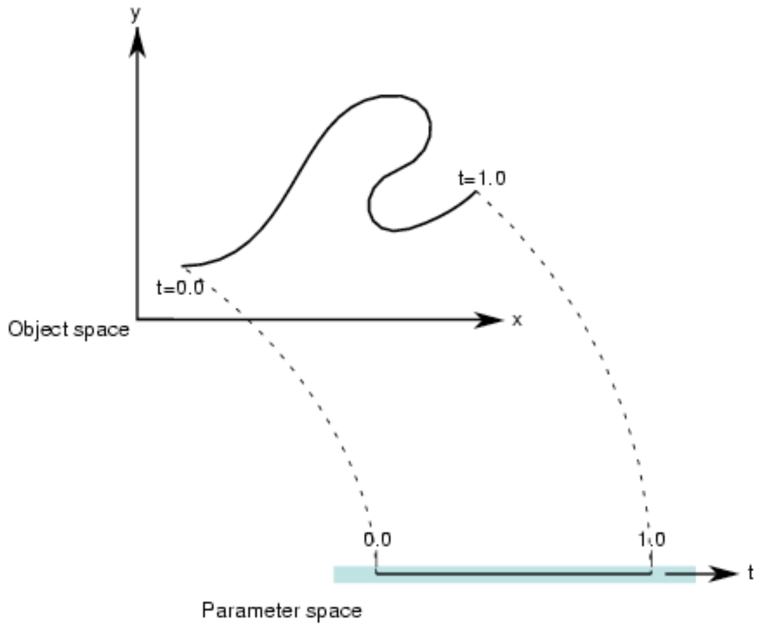
Parametric Representation

- The contour itself is a given in parametric form

$$c(s) = (x(s), y(s))$$

where $x(s)$ and $y(s)$ are the coordinates along the contour and $s \in [0,1]$





Energy Functional

- The contour $c(s)$ is deformed using the sum of the three forces $E_{cont}, E_{curv}, E_{img}$
- How? We construct an *energy functional which measures the appropriateness* of the contour.

$$\mathcal{E} = \int \left(\alpha(s) E_{cont} + \beta(s) E_{curv} + \gamma(s) E_{img} \right) ds$$

where α, β and γ control the relative influence of the corresponding energy terms and can vary along *c.*

- Good solutions correspond to *minima of the functional.*
- Goal: minimize this functional with respect to the contour parameter *s*.

Continuity Term

- The continuity term, E_{cont} , encourages continuity of the contour and is defined as:

$$E_{cont} = \left\| \frac{dc}{ds} \right\|^2$$

- It is based on the 1st derivative. For a continuous curve we want to minimize E_{cont}
- It is a form of an internal energy function.
- The term internal energy is used to refer to how rigid, stiff the deformable model is.
- The goal of an internal energy function is to:
  - enforce a shape on the deformable contour and
  - maintain a constant distance between the points in the contour.

Continuity Term- Discrete Case

- In the discrete world the contour is replaced by a chain of *N* image points on the curve, $p_1, p_2, \ldots, p_N$
- The first derivative is then approximated by a finite difference:

$$E_{cont} = \left\| p_i - p_{i-1} \right\|^2 \quad \text{where} \quad i = 2,3,\ldots,N$$

$$E_{cont} = (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2$$

- Thus, this term tries to minimize the distance between the points. It supports more compact contours.

Continuity Term – A Better Approximation

$$E_{cont} = \left\| p_i - p_{i-1} \right\|^2$$

- As defined, E_{cont} can cause the formation of clusters.
- Thus, a better form is:

$$E_{cont} = \left(\bar{d} - \left\| p_i - p_{i-1} \right\| \right)^2 \quad \text{where} \quad \bar{d} = \frac{1}{N-1} \sum_{i=2}^{N} \left\| p_i - p_{i-1} \right\|$$

- When $\left\| p_i - p_{i-1} \right\| >> \bar{d}$ then $E_{cont} \approx \left\| p_i - p_{i-1} \right\|^2$. However if we don't have such outliers, i.e. for smaller distances this new E_{cont} encourages the formation of equally spaced chains of points.

Continuity Term - Comments

- In the absence of other influences, the continuity energy term coerces:
  - an open deformable contour into a straight line and
  - a closed deformable contour into a circle.

Smoothness Term

- The smoothness term, E_{curv} , encourages smoothness of the contour and is defined as:

$$E_{curv} = \left\| \frac{d^2 c}{ds^2} \right\|^2$$

- It is based on the 2nd derivative, which is a measure of curvature.
- We want to avoid oscillations => Penalize high curvature.
- Thus, for a smooth curve we want to minimize E_{curv}
- It is also a form of an internal energy function. In this case, itenforces a particular shape preference (smooth shapes).

Smoothness Term- Discrete Case

- Since the contour is replaced by a chain of *N* image points on the curve, $p_1, p_2, \ldots, p_N$, the second derivative is again approximated by a finite difference:

$$E_{curv} = \left\| p_{i+1} - 2p_i + p_{i-1} \right\|^2 \quad \text{where} \quad i = 2,3,\ldots,N-1$$

$$E_{curv} = (x_{i+1} - 2x_i + x_{i-1})^2 + (y_{i+1} - 2y_i - y_{i-1})^2$$

Edge Attraction Term

- The edge attraction term, E_{img} , attracts (pulls) the contour towards an edge-defined target contour and is defined as:

$$E_{img} = -\left\|\nabla I\right\|$$

  where ∇I is the spatial gradient of the intensity image I, computed at each contour point.
- At large gradient vectors (i.e. close to the image edges) we obtain very small (negative) E_{img} values.
- It is a form of an external energy function.

Energy Functional- Revisit

- Recall that in order to deform a curve $c(s)$ so that it closely matches the target curve, we minimize the energy functional:

$$\mathcal{E} = \int \left(\alpha(s) E_{cont} + \beta(s) E_{curv} + \gamma(s) E_{img} \right) ds$$

- $\mathcal{E}$ is minimal when each of the three forces is minimal, which means:
  - E_{cont} forces a compact curve (prefers lines and circles)
  - E_{curv} avoids oscillations (ridges).
  - E_{img} is small when the active contour is close to the edge.

Energy Functional- Discrete case

- Since the contour is replaced by a chain of *N* image points on the curve, $p_1, p_2, \ldots, p_N$ we need a discrete approximation to the energy functional:

$$\mathcal{E} = \sum_{i=1}^{N} \alpha_i E_{cont} + \beta_i E_{curv} + \gamma_i E_{img}$$

where $\alpha_i, \beta_i, \gamma_i \geq 0$

- Typical values for the weighting parameters are:

$$\alpha_i = \beta_i = \gamma_i = 1, \text{ or } \quad \alpha_i = \beta_i = 1 \text{ and } \gamma_i = 1.2$$

Last Step: Minimization

- So computing an active contour involves setting up an energy functional like

$$\mathcal{E} = \sum_{i=1}^{N} \alpha_i E_{cont} + \beta_i E_{curv} + \gamma_i E_{img}$$

  and minimizing it.
- There are many different ways to solve this optimization problem.
- One of the most efficient methods (when applicable) for solving optimization problems is greedy algorithms (looks at locally optimal solution and that leads to a globally optimal solution).

Greedy Algorithm

1. **Greedy Minimization**: Move each point p_i within a small neighborhood to the pointthat minimizes the functional. Do computations over a small neighborhood: 3x3 or 5x5. Compute the energy at each location in the neighborhood and pick the smallest one. Call this smallest one p_i'.
2. **Corner Elimination**: Look for corners among all the p_i' and adjust β_i to smooth them out. Corners, if present should have the largest curvature values. If a point p_j' has the largest E_{curv} value, then set β_j=0. This way we neglect the contribution of E_{curv} at point p_j' and let the other terms move the contour.
3. Go back to step 1, until a predefined number of points reaches a local minimum.

Greedy Algorithm Details

- E_{cont}, E_{curv} and E_{img} must be normalized.
- For E_{cont} and E_{curv} we divide by the largest value in the neighborhood in which the point can move.
- For E_{img}, let M and m be the maximum and minimum values of ∇I over the neighborhood. We normalize then by:

$$E_{img} = -\frac{\|\nabla I\| - m}{M - m}$$

Greedy Algorithm - Comments

- Typically the number of iterations until convergence is proportional to the number of points on the contour, e.g. 4* (# points).
- It has low computational requirements *O(MN).*
- It works well when the initial contour is close to the target contour.
- There is no guarantee of convergence to the global minimum.

Snake Algorithm

Let f be the *minimum fraction* of points that must move in each iteration before convergence, i.e. if fewer than f points points moved, then the deformable contour has stabilized to its final shape.

While a fraction greater than f of snake points move in an iteration:

1. For each i =1 to N

   a. compute $\mathcal{E}$ for each point in the 3x3 neighborh.

   b. find the location in the neighborh. Where $\mathcal{E}$ is min. and move p_i at that location.

Snake Algorithm -continued

2. For each i =1 to N

    a. compute $k = \left\| p_{i+1} - 2p_i + p_{i-1} \right\|$

    b. find max k and all locations where k>threshold

    c. let p_j be the point with max k

    d. set β_j= 0

3. update average distance *d*, *d_bar*.

Return the chain of points p_j that represent the deformable contour.

Image Sources

1. Movies on active contours are courtesy of. C. Xu and J. Prince http://www.iacl.ece.jhu.edu/static/gvf/
2. The and-drawn parametric curve is courtesy of G. Bebis, http://www.cse.unr.edu/~bebis/CS791E/Notes/DeformableContours.pdf
3. The image of the parametric curve, together with the parameter space is courtesy of sgi, http://techpubs.sgi.com/library/dynaweb_docs/0650/SGI_Developer/books/Perf_PG/sgi_html/figures/parametric.curve.gif